

00000000000000000000000000000000

## BACKGROUND OF THE INVENTION

## TI-29602 Page 1

CDMA code for each user is typically produced as the modulo-2 addition of a Walsh code with a pseudo-random code (two pseudo-random codes for QPSK modulation) to improve the noise-like nature of the resulting signal. A cellular system as illustrated in Figure 4 could employ IS-95 or WCDMA for the air interface between the base station and the mobile user station.

A spread spectrum receiver synchronizes with the transmitter by code acquisition followed by code tracking. Code acquisition performs an initial search to bring the phase of the receiver's local code generator to within typically a half chip of the transmitter's, and code tracking maintains fine alignment of chip boundaries of the incoming and locally generated codes. Conventional code tracking utilizes a delay-lock loop (DLL) or a tau-dither loop (TDL), both of which are based on the well-known early-late gate principle.

In a multipath situation a RAKE receiver has individual demodulators (fingers) tracking separate paths and combines the results to improve signal-to-noise ratio (SNR), typically according to a method such as maximal ratio combining (MRC) in which the individual detected signals are synchronized and weighted according to their signal strengths. A RAKE receiver usually has a DLL or TDL code tracking loop for each finger together with control circuitry for assigning tracking units to received signal paths. Figure 5 illustrates a receiver with N fingers.

The UMTS (universal mobile telecommunications system) approach UTRA (UMTS terrestrial radio access) provides a spread spectrum cellular air interface with both FDD (frequency division duplex) and TDD (time division duplex) modes of operation. UTRA currently employs 10 ms duration frames partitioned into 15 time slots with each time slot consisting of 2560 chips. In FDD mode the base station and the mobile user transmit on different frequencies, whereas in TDD mode a time slot may be allocated to transmissions by either the base station (downlink) or a mobile user (uplink). In addition, TDD systems are differentiated from the FDD systems by the presence of interference cancellation at the receiver. The spreading gain for TDD

systems is small (8-16), and the absence of the long spreading code implies that the multi-user multipath interference does not look Gaussian and needs to be canceled at the receiver.

In currently proposed UTRA a mobile user performs an initial cell search when first turned on or entering a new cell; this search detects transmissions of base stations on the physical synchronization channel (PSCH) without any scrambling. The initial cell search by a mobile user must determine timing (time slot and frame) plus identify pertinent parameters of the found cell such as scrambling code(s).

For FDD mode the physical synchronization channel appears in each of the 16 time slots of a frame and occupies 256 chips out of the 2560 chips of the time slot. Thus a base station transmitting in the synchronization channel a repeated primary synchronization code of pseudo-noise of length 256 chips modulated by a length 16 comma-free code (CFC) allows a mobile user to synchronize by first synchronizing to the 256-chip pseudo-random code to set slot timing and then using the cyclic shift uniqueness of a CFC to set frame timing. Further, decoding the CFC by the mobile user reveals the scrambling code used by the base station.

In contrast, for TDD mode the physical synchronization channel only appears in one or two time slots per frame, so the length-16-CFC-modulated primary synchronization code does not easily apply. An alternative proposed TDD mode initial cell search employs a sum of a primary synchronization code (PSC) plus six secondary synchronization codes (SSCs); each code is a 256-chip pseudo-noise sequence, and the codes are orthogonal. In this proposal the initial cell search consists of slot synchronization, frame synchronization and code group identification with scrambling code determination. In particular, during slot synchronization the mobile user employs the PSC to acquire slot synchronization to the strongest cell (strongest received base station transmission); the PSC is common to all cells. A single matched filter (or any similar device matched to the PSC) may be used for detection.

Next, the mobile user employs the six SSCs to find frame synchronization and to identify one out of 32 code groups being used by the base station. Each of the six SSCs is modulated by +1 or -1; this implies 5 bits of information to identify which one of 32 possible code groups is used by the found base station (scrambling codes and midambles), and the sixth SSC is modulated by +1 or -1 to identify whether the time slot is the first or second physical synchronization channel slot in the frame (frame synchronization). Each of the six SSCs is scaled by  $1/\sqrt{6}$  to make the power of the sum of the six modulated SSCs equal to the power of the PSC. Lastly, the mobile user determines which of the four scrambling codes in the cell's code group is being used by, for example, correlation on the common control physical channel.

However, this TDD mode proposal has problems including low signal to noise ratio in the sum of the six modulated SSCs.

There is also a problem of efficient decoding within TDD mode proposals.

#### SUMMARY OF THE INVENTION

The present invention provides TDD mode cell search with comma-free codes from an alphabet of sums of modulated secondary synchronization codes. Preferred embodiments use length 2 or 4 comma-free codes for frames with two time slots for a synchronization channel and non-interleaved or two-level interleaved, respectively.

This has advantages including increased signal-to-noise ratios due to the removal of redundancy.

#### BRIEF DESCRIPTION OF THE DRAWINGS

The drawings are heuristic for clarity.

Figures 1a-1b show synchronization channel and comma-free codes.

Figure 2 shows a spread spectrum system.

Figures 3a-3b illustrate pseudo-random code and symbols.

Figures 4-5 show a cellular system plus a block diagram of a receiver.

Figures 6-7b illustrate comma-free codes.

Figures 8-15 are simulation results.

008280" 06E64960

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

### 1. Overview

Preferred embodiment synchronization methods for UTRA-type TDD mode initial cell search use length 2 or 4 (or more) comma-free codes (CFCs) to encode both frame timing and base station code group information plus frame position for interleaved frames. The use of CFCs effectively removes the redundancy of straightforward coding of frame timing and code group in both time slots occupied by the physical synchronization channel. Some preferred embodiments use CFCs with alphabets generated from scaled sums of two or three QPSK-modulated secondary synchronization codes added to a primary synchronization code. Preferred embodiment spread spectrum communication systems incorporate preferred embodiment synchronization methods.

In preferred embodiment communications systems the base stations and the mobile users could each include one or more digital signal processors (DSP's) and/or other programmable devices with stored programs for performance of the signal processing of the preferred embodiment synchronization methods. The base stations and mobile users may also contain analog integrated circuits for amplification of inputs to or outputs from antennas and conversion between analog and digital; and these analog and processor circuits may be integrated on a single die.. The stored programs may, for example, be in ROM onboard the processor or in external flash EEPROM. The antennas may be parts of RAKE detectors with multiple fingers for each user's signals. The DSP core could be a TMS320C6x or TMS320C5x from Texas Instruments.

### 2. Preferred embodiments with 12 secondary synchronization codes

First preferred embodiment initial cell search methods by mobile users in a UTRA system in TDD mode employ the physical synchronization channel as illustrated in Figure 1a together with two-level frame interleaving. In particular, Figure 1a shows a 10 ms frame, 15 time slots per frame with 2560 chips per slot; the physical

[illegible]

Frames may be interleaved (typically to a depth of 2 or 4) in order to mitigate burst noise. The preferred embodiment CFC has codeword length equal to the number of physical synchronization channel time slots per frame multiplied by the frame interleave depth.

TI-29602 Page 7







Step 2. Frame synchronization and code-group identification plus frame position  
 During the second step the mobile user employs the secondary synchronization codes (SSCs) to find frame synchronization and identify one out of 32 code groups. Each code group is linked to a time offset of the synchronization channel within the slot, thus to a specific frame timing, and also is linked to a set of four scrambling codes (and basic midambles). To detect the position of the next synchronization slot, correlate PSC with the received signal at both 7 and 8 time slots after the slot detected in step 1. (The 7 and 8 arises because frames have 15 time slots.) The received signal at the positions of the synchronization slots is correlated with the PSC and all of the SSCs. These correlations may be done coherently over one or many time slots with phase correction provided by the correlation with PSC. The correlations recover the set of three  $b_i$  modulations for a time slot. Lookup in the table of Figure 1b yields the code group, time offset of the synchronization channel within the time slot, and the frame timing and interleave position.

### Step 3. Scrambling code identification

During the third step the mobile user determines which of the four scrambling codes (and basic midambles) in the code group the cell is actually using. This may be by, for example, correlations with all four scrambling codes on the common control channel transmissions by the base station.

Preferred embodiment mobile users would have the PSC and SSCs (and scrambling codes and so forth) stored in memory and be programmed to execute the foregoing cell search.

### 3. Code minimum distance

The correlations in step 2 to find the QPSK modulation symbols  $\{b_i\}$  are limited by signal-to-noise factors which relate to the minimum distance between possible codeword components being detected. The general minimum distance between two sums of three SSCs with QPSK modulation (e.g.,  $\sum b_i c_i / \sqrt{3}$  and  $\sum b'_i c_i / \sqrt{3}$ ) occurs





00000000-00000000

A	B	C
A	B	C
A	C	B
B	C	A.

TI-29602 Page 13

```

C1 = nchoosek(C(1,:),2);
C1 = [C1 flipud(C(1,:))];
C1 = [C1(1,:); C1];
for k=2:32
    Ck = nchoosek(C(k,:),2);
    Ck = [Ck flipud(C(k,:))];
    Ck = [Ck(1,:); Ck];
    C1 = [C1; Ck];
end
else
    error('Only 256 group case currently');
end

```

```

snr_list = 10.^(snr_list/10);

```

```

switch(fd)
case 0
    rayl_ch = ones(1,20);
case 5
    load /db/wcdma/200_Hz/fading_5_hz/old_sig_1.mat
    rayl_ch = old_sig_1.';
    clear old_sig_1;
    ch_len = length(rayl_ch);
case 20
    load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_20/rayl_sig
case 40
    load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_40/rayl_sig
case 80
    load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_80/rayl_sig
case 100
    %load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_100/rayl_sig
    %load

```

```

%/db/wcdma/32_KSPS/fading_1000_hz/NON_POW_CTRL/non_pow_ctrl_data_1_1
.mat
load /db/wcdma/200_Hz/fading_100_hz/old_sig_1.mat
%rayl_ch = old_sig_1(1:16:500000).';
rayl_ch = old_sig_1.';
clear old_sig_1;
ch_len = length(rayl_ch);
%load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_100/rayl_sig
case 460
    load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_460/rayl_sig
case 1000
    rayl_ch = ones(1,20);

```

```

otherwise
    error('Unknown Doppler Frequency');
end

if (fd~=0)&(fd ~= 5)&(fd ~=100)
    ch_len = length(rayl_ch);
    rayl_ch = rayl_ch(:,1:20:ch_len); % subsample to get it at 16kps
    ch_len = length(rayl_ch);
end

sc_rat = 0.5;
ssc_rat = sqrt(sc_rat);

pd_list=[];

tdec = [];
tpr = [ones(1,ceil(num_slots/4)); -ones(1,ceil(num_slots/4))];
tpr = reshape(tpr,1,prod(size(tpr)));
tpr = repeat(tpr,2);

eval(['load cr_corr_' int2str(q) '_' int2str(foff) 'khz.mat']);
pp = psc(1);
pscc = psc(2:q+1,:);
psc = psc.';

for snr = snr_list
    snr
    pd=0;
    ssc_rat_sc = sqrt(snr)*ssc_rat;
    for iter = 1:max_iter
        index = ceil(rand*ng);
        if index == 0
            index = 1;
        end

        cyc = ceil(rand*4);
        if cyc == 0
            cyc = 1;
        end

        %index = 1;
        %cyc=2;

        if (fd ~= 0)&(fd ~=1000)&(fd~=5)&(fd~=100)
            start = ceil(rand*(ch_len-8*num_slots));
        end
    end
end

```

```

if start == 0
    start = 1;
end
fading = rayl_ch(1,start:8:start+8*num_slots-1);
elseif (fd==5)|(fd==100)
    start = ceil(rand*(ch_len-num_slots));
    if start == 0
        start = 1;
    end
    fading = rayl_ch(1,start:start+num_slots-1);
elseif fd == 0
    fading = ones(1,num_slots);
elseif fd == 1000
    fading = (randn(1,num_slots)+j*randn(1,num_slots))/sqrt(2);
end

%C(index,:);
%size(psc)

k1 = ceil(index/2);
k2 = mod((index-1),2);
k3 = mod(k1-1,4);

tp = 1-2*bin_state(k2,2);

tv = ssc(:,C1(k1,:));
tva = tv(:,[1,2]).*tp(ones(q,1),:);
if k3
    tva = tva*j;
end
tva = sum(tva,2);
tvb = tv(:,3);

%disp('heh')
out_var = zeros(q,num_slots);
tv = zeros(q,4);

tpr1 = [tpr(cyc:length(tpr)) tpr(1:cyc-1)];
tpr1 = tpr1(1:num_slots);
for nk = 1:4
    ttem = tpr1(nk)*tva+((-1)^(nk+cyc))*tvb;
    out_var(:,nk:4:num_slots) = ttem(:,ones(1,num_slots/4));
end

tv = ssc rat sc*fading;

```



0964930 : 058600

00000000000000000000000000000000

```
%eval(['save pd_list cfcn ' int2str(fd) 'hz_coh pd_list snr_list']);
```

TI-29602 Page 18

the subscript offset of 1: codes numbered 1-16 in the simulation correspond to codes  $C_0, \dots, C_{15}$  of Figures 7a-7b.)

Let  $N(m,n)$  be the correlation of the  $m$ th synchronization code with the received (simulated) signal in the  $n$ th detected time slot of the synchronization channel. The preferred embodiment decoding defines  $N_1$ ,  $N_2$ , and  $N_3$  as the column vectors of sums and differences of correlations over the detected time slots as follows:

Include any correlations for time slot numbers greater than 4 by treating  $N(m,n)$  as  $N(m,n \bmod 4)$ . Thus  $N_1$ ,  $N_2$ ,  $N_3$  are just differing combinations of the correlations over the slots. Note that Figure 7a illustrates the significance of these  $+$   $-$  patterns in the definitions of  $N_1$ ,  $N_2$ , and  $N_3$ ; namely, for each code group slot  $k$  and slot  $k+8$  of frame 1 (corresponding to successive slots in the simulation program) have signs for the first two codes opposite those of slots  $k$  and  $k+8$  of frame 2 (which correspond to the next two successive slots in the simulation program). Thus  $N_1$  corresponds with frame alignment and  $N_2$  with a shift of one slot. Further, the third code has opposite signs in slots  $k$  and  $k+8$  in both frames which  $N_3$  tracks. That is, if the received signal used a code group of just the  $q$ th code with signs  $++--$ , then  $N_1(q)$  will roughly be as follows:  $+4$  if the first detected time slot is slot  $k$  of frame 1,  $0$  if the first detected time slot is slot  $k+8$  of frame 1,  $-4$  if the first detected time slot is slot  $k$  of frame 2, and  $0$  if the first detected time slot is slot  $k+8$  of frame 2. Similarly,  $N_2(q)$  will roughly be  $0$  if the first detected time slot is slot  $k$  of frame 1,  $+4$  if the first detected time slot is slot  $k+8$  of frame 1,  $0$  if the first detected time slot is slot  $k$  of frame 2, and  $-4$  if the first detected time slot is slot  $k+8$  of frame 2.



corresponding decoding method adjustment while retaining the formation of matrices of correlations with phase adjustments ( $\pm$  and  $j$  factors) combined to form all of the code group correlation sums followed by search for the maximum which yields the cyclic shift and code group indication.

In a more general encoding there could be  $q$  synchronization codes and each set of three codes (a code set) gives rise to 64 sums of QPSK-modulated codes (that is, each of the three codes in a code set can have one of 4 modulations:  $+1$ ,  $-1$ ,  $+j$ ,  $-j$ , so there are  $4^3$  combinations). Thus for length four CFCs with such sums as elements without repetition, there will be 16 possible code groups (4 sums in each code group), and further restriction to code groups with only an even number of  $\pm j$  modulations in the sums halves the number of code groups to 8 per code set. Similarly, for length 2 CFCs there will be 32 code groups per code set, and further restriction to an even number of  $\pm j$  modulations in the sums halves the number to 16 code groups. In fact, Figure 6 illustrates  $q=6$ , length 2 CFCs with two code sets and 32 code groups.

The CFC elements are detected in  $n$  time slots (if  $n$  is greater than the length of the CFC, then average over the detected time slots modulo the CFC length). Preferred embodiment decoding proceeds with steps such as:

(1) detect the correlations of each of the  $q$  synchronization codes with the received baseband signal in each of the  $n$  time slots; this yields  $q \cdot n$  correlations for  $n$  less than or equal to 2 or 4 (CFC lengths). For  $n$  greater than 2 or 4 average over time slots modulo 2 or 4 to have  $q \cdot 2$  or  $q \cdot 4$  correlations. Note that the array  $N$  in the foregoing simulation corresponds to these correlations prior to any averaging modulo 4. Denote the correlation with  $A$  of the received baseband signal in the first detected time slot (modulo 2 or 4) as  $N(A,1)$ , the correlation with  $C$  in the second detected time slot with  $C$  as  $N(C,2)$ , and so forth. For clarity, first consider length 4.

(2) form signed sums of these correlations. That is, define arrays N1, N2, and N3 as sums over the four time slots with two positive signs and two negative signs:

$$N1(A) = N(A,1) + N(A,2) - N(A,3) - N(A,4)$$

$$N_2(A) = N(A,1) - N(A,2) - N(A,3) + N(A,4)$$

$$N_3(A) = N(A,1) - N(A,2) + N(A,3) - N(A,4)$$

And similarly for other synchronization codes.

(3) Then define arrays  $Nt1$ ,  $Nt2$ , and  $Nt3$  which are row-indexed by the code groups used in the CFC and with entries corresponding to the  $N1$ ,  $N2$ , and  $N3$  with multipliers for the codes of the row's code group. For example, consider a code set of the three synchronization codes  $A$ ,  $B$ , and  $C$ , which leads to 8 code groups including  $\{A+B+C, A+B-C, -A-B+C, -A-B-C\}$ ,  $\{jA+jC+B, jA+jC-B, -jA-jC+B, -jA-jC-B\}$ , and so forth. (Such a set of 8 code groups corresponds to 8 adjacent lines in Figure 7a.) Thus define  $Nt1$  and  $Nt2$  as arrays with two columns (for the first and second codes in a code group) and  $Nt3$  as an array with one column (for the third code in the code group):

$$\text{Nt1}(\{A+B+C, A+B-C, -A-B+C, -A-B-C\}, [1,2]) = [\text{N1}(A), \text{N1}(B)]$$

$$Nt2(\{A+B+C, A+B-C, -A-B+C, -A-B-C\}, [1,2]) = [N2(A), N2(B)]$$

$$Nt_3(\{A+B+C, A+B-C, -A-B+C, -A-B-C\}) = N_3(C)$$

and

$$\text{Nt1}(\{jA+jC+B, jA+jC-B, -jA-jC+B, -jA-jC-B\}, [1,2]) = [-jN1(A), -jN1(C)]$$

$$\text{Nt2}(\{ |A\rangle + |C\rangle + |B\rangle, |A\rangle + |C\rangle - |B\rangle, -|A\rangle - |C\rangle + |B\rangle, -|A\rangle - |C\rangle - |B\rangle \}, [1,2]) = [-jN_2(A), -jN_2(C)]$$

$$Nt3(\{jA+jC+B, jA+jC-B, -jA-jC+B, -jA-jC-B\}) = N3(B)$$

That is, for Nt1 and Nt2 the entry for a row is the pair of N1s and N2s corresponding to the two codes which maintain relative modulations within the code group; additionally, apply multipliers to make the first sum of the code group have positive real modulations for these two codes. The Nt3 entry is the N3 for the third code of the code group; this does not need a multiplier because the first sum of the code group always has a +1 modulation for the third code.

(4) for each code group form the 4 sums (which correspond to the first detected time slot falling into one of the four time slots of the length 4 CFC) of the components of  $N_{t1}$  (or  $N_{t2}$ ) plus  $N_{t3}$ . That is, code group  $\{A+B+C, \dots\}$  leads to the four sums

$$N1(A) + N1(B) + N3(C)$$

$$N_2(A) + N_2(B) - N_3(C)$$

$$-N1(A) - N1(B) + N3(C)$$

-N2(A) - N2(B) - N3(C)

and the code group  $\{jA+jC+B, \dots\}$  has the sums:

$$\begin{aligned} & -jN1(A) - jN1(C) + N3(B) \\ & -jN2(A) - jN2(C) - N3(B) \\ & jN1(A) + jN1(C) + N3(B) \\ & jN2(A) + jN2(C) - N3(B) \end{aligned}$$

(5) find the maximum of the sums of step (4). The code group is that yielding the maximum, and the cyclic shift (the slot number of the first detected slot) is the number in the set of the four sums.

#### 7. Preferred embodiments without frame interleaving

The fourth preferred embodiments apply to synchronization with non-interleaved frames but with 2 (or more) time slots per frame occupied by the physical synchronization channel. In particular, with  $C_1, C_2, \dots, C_6$  being six SSCs, length 2 CFC codewords using QPSK modulation can be formed:  $\langle (C_i + C_k)/\sqrt{2}, (C_i - C_k)/\sqrt{2} \rangle$ ,  $\langle -(C_i + C_k)/\sqrt{2}, -(C_i - C_k)/\sqrt{2} \rangle$ ,  $\langle j(C_i + C_k)/\sqrt{2}, j(C_i - C_k)/\sqrt{2} \rangle$ , and  $\langle -j(C_i + C_k)/\sqrt{2}, -j(C_i - C_k)/\sqrt{2} \rangle$  where  $C_i$  and  $C_k$  are a pair selected from the set  $C_1, C_2, \dots, C_6$  without replacement. With six SSCs there are 15 pairs  $\{C_i, C_k\}$  and thus 60 codewords of the foregoing type.

For the case of two time slots per (non-interleaved) frame and 32 code groups to identify with detection in a single time slot, using six SSCs with BPSK modulation and scaled by  $\sqrt{6}$  as previously described, the minimum code distance between two possible detections is  $2/\sqrt{6} = 0.816$ . In contrast, with the foregoing length 2 CFC codewords the frame timing is inherent in the codeword component ( $C_i + C_k$  is the first time slot and  $C_i - C_k$  the second time slot) and the minimum code distance between two possible detections equals 1. Note that 30 of the 32 codewords needed can be with real (in phase) modulation and only two codewords need imaginary (quaternary) modulation. That is, 30 codewords may be of the form:  $\langle (C_i + C_k)/\sqrt{2}, (C_i - C_k)/\sqrt{2} \rangle$  and  $\langle -(C_i + C_k)/\sqrt{2}, -(C_i - C_k)/\sqrt{2} \rangle$ , and only two imaginary modulated codewords need be used, for example,  $\langle j(C_1 + C_2)/\sqrt{2}, j(C_1 - C_2)/\sqrt{2} \rangle$  and  $\langle -j(C_1 + C_2)/\sqrt{2}, -j(C_1 - C_2)/\sqrt{2} \rangle$ . And reflecting the greater minimum code distance, Figures 8-15 illustrate

the superior performance of such length 2 CFC codes versus the BPSK-modulated six SSCs.

Further, the computational complexity of the length 2 CFC is comparable to that of the six BPSK-modulated SSCs. In particular, for the six SSCs, 32 correlations are obtained performing length 8 correlations and the Fast Hadamard Transform is applied to these correlation values to obtain the correlations with the 6 SSCs and the PSC, requiring  $8 \times 32 + 2 \times 16 \times \log_2 16 + 7 = 391$  complex additions. Again, the phase of the correlation with the PSC is used as a reference for the correlations with the 6 SSCs. This requires 28 real multiplications and 13 real additions. There are 64 possible combinations (codewords) of the 6 SSCs. Noting that some of these combinations are simply negatives of others and using other redundancies, the 64 combinations require 564 real additions. Averaging the 64 decision variables over K slots requires approximately 64 real additions per slot for large K. Selecting the maximum after averaging over K slots requires  $(\log_2 64)/K$  compares per slot. This number is the same for both the CFC and the six SSC methods and for large K is very small. So neglect it in the analysis. Thus the six modulated SSC method requires 118 real additions per slot for computing, averaging the decision variables and selecting the maximum.

For the CFC, the multiplication by j is imply flipping the imaginary and real parts. Also, as noted above using  $\pm \angle (C_i + C_k)/\sqrt{2}, (C_i - C_k)/\sqrt{2} >$  produces 30 codewords so only  $\pm \angle (C_1 + C_2)/\sqrt{2}, (C_1 - C_2)/\sqrt{2} >$  are needed to complete the 32 required codewords. Again noting that some of these combinations are simply negatives of the others, the method requires 32 additions per slot to obtain the 64 decision variables. Averaging the 64 decision variables over K slots requires approximately 64 real additions per slot for large K. Again neglecting the compares required for selecting the maximum, which is the same for both methods, the length 2 CFC method requires 96 real additions per slot for computing, averaging the decision variables, and selecting the maximum.



## 8. Modifications

The preferred embodiments may be modified in various ways while retaining the features of comma-free codes (CFC) for synchronization in TDD systems. For example, the modulations  $b_i$  can be chosen from the set  $\{e^{j\theta} e^{j\theta+j\pi/2} e^{j\theta+j\pi} e^{j\theta+j3\pi/2}\}$ . In the preferred embodiment choosing  $b_i$  from the set  $\{+1 +j -1 -j\}$  is equivalent to setting  $\theta=0$ .

Also, if the length-4 comma free code from the preferred embodiment is represented as  $\{s_1 s_2 s_3 s_4\}$  where  $s_k = \sum b_i c_i / \sqrt{N}$ , any permutation of this code is also a comma free code, e.g.  $\{s_2 s_4 s_3 s_1\}$ , which is one such permutation, is also a comma free code. This implies that one can achieve the same comma free code performance by transmitting  $\{s_2 s_4 s_3 s_1\}$  in sequence instead of  $\{s_1 s_2 s_3 s_4\}$ .

And for two length-4 comma free codes from the preferred set, say  $\{s_1 s_2 s_3 s_4\}$  and  $\{g_1 g_2 g_3 g_4\}$ , then the codes formed by swapping any two elements between the two code words also form two length-4 CFC's, e.g.  $\{s_1 g_2 s_3 s_4\}$  and  $\{g_1 g_3 s_2 g_4\}$  are also valid comma free codewords which can be used instead of  $\{s_1 s_2 s_3 s_4\}$  and  $\{g_1 g_2 g_3 g_4\}$ . Swapping the elements between three codewords forms a new set of three comma free codewords.

Further, using the construction method for CFCs made from two or three parallel SSCs, one can construct other CFC's with four or more SSC's in parallel.

Analogously, one can easily increase the length of the comma free codewords by concatenating two comma free codewords or portions of two comma free codewords. Similarly splitting the comma free codewords can also decrease the length.

Note that the use of comma-free codes for TDD permits all time slots (of the synchronization channel) contribute to the distance between codewords to avoid the problem of loss of diversity leading to a loss of codeword separation distance.